I.J.E.M.S., VOL.4 (3) 2013: 326-331 ISSN 2229-600X

INTERNATIONAL JOURNAL OF ENGINEERING AND MANAGEMENT SCIENCES

www.scienceandnature.org

QUERY OPTIMIZATION ALGORITHM BASED ON RELATIONAL ALGEBRA EQUIVALENCE TRANSFORMATION

Seema, Parminder Kaur
Department of Computer Science & Engineering, Guru Nanak Dev University, Amritsar, Punjab, India

ABSTRACT
Query optimization is an important part of database management system. The objective of query optimization is to find an execution strategy for a distributed query which can be defined with relational algebra operations and communication primitives for transferring data between sites.This research paper suggests the heuristic approach for selecting the optimal evaluation plan and Semi-join approach for reducing the communication cost. Calculating the cost of each evaluation plan of a query takes lots of computational efforts as well as time. The communication cost in distributed database is mainly depending upon the amount of data transferred from one site to another site. In relational algebra expression, the most time and space operation is Cartesian product with the help of join operations. This paper performs the implementation of selection and projection operations in the relational algebra expression in order to avoid the direct Cartesian product operation. A series of selection and projection operations are applied on an example query to reduce the size of intermediate relations and thus achieve the target of query optimization. A simulation study is also performed to show that semi-join operation is more preferable to reduce the communication cost.

KEYWORDS: Packet Filtering, Stateless, Application, Proxies,

INTRODUCTION

Distributed database (DDB) is defined as a collection of multiple, logically interrelated databases distributed over a computer network (Ozsu and Valduriez, 1991). The databases within the distributed database could be distributed not only across the local area network inside one company but also across several companies through a wide area network. As with a traditional centralized database, distributed database systems must hide the heterogeneity of the data resources and provide a unified way to access and manage those data resources.

Distributed database management system (DDBMS) provides the functions to manage the distributed database and makes the distribution transparent to users. It helps users to manipulate the entire distributed database without considering the underlying data distribution details (Ghaemi and Milanifard, 2008).

Query Optimization refers to the process by which the best execution strategy for a given query is found from a set of alternatives. The goal of query optimization is to find an execution strategy for the query that is close optimal. An execution strategy for a distributed query can be described with relational algebra operations and communication primitives (send/receive operations) for transferring data between sites (Oszu and Valduriez, 1997).

RELATED WORK

(Bernstein and Chiu, 1981) have used semi-joins to solve relational queries. The semi-join is a relational algebraic operation that selects a set of tuples in one relation that match one or more tuples of another relation on the joining domains. The results show that for tree queries, as soon as a strategy embeds , a full reduction and hence the full potential of semi-joins are achieved. Furthermore, tree-query membership can be tested in linear time. This suggests that searching for optimal strategies is quite likely to be easier for tree queries than for cyclic ones. For cyclic queries, finding good semi-join programs is likely to be quite difficult.

(Hevner et.al, 1983) have suggested a new algorithm GENERAL to derive processing strategies for arbitrarily complex queries. Three versions of the algorithm are given: one for minimizing response time and two for minimizing total time. The algorithm is shown to provide optimal solutions under certain conditions. To minimize response time of a processing strategy, parallel data transmissions are emphasized by the use of Algorithm PARALLEL and Procedure RESPONSE. To minimize the total time of a processing strategy, serial time transmissions are emphasized by the use of Algorithm SERIAL and Procedure TOTAL in Algorithm GENERAL.

(Talbot, 1984) have presented several techniques which optimize a relational database interface and have attempted to isolate the physical aspects of optimization in order to present a portable logical optimizer. The techniques that have been presented and illustrated in the functional design of a logical optimizer, including both old and new methods for optimization are: algebraic transformations, Common-subexpression recognition, Null relation removal and removal of superfluous joins, Identification and storage of subexpressions over the query mix.

(Chung and Irani, 1986) have addresses the processing of a query in distributed database systems using a sequence of semijoins. The objective is to minimize the intersite data traffic incurred by a distributed query. A method is developed which accurately and efficiently estimates the size of an intermediate result of a query. This method provides the basis of the query optimization algorithm. Since the distributed query optimization problem is known to be intractable, a heuristic algorithm is developed to determine a low-cost sequence of semijoins. The cost comparison with an existing algorithm is provided. The complexity of the main features of the algorithm is analytically derived. The scheduling time for sequences of semijoins is measured for example queries using the PASCAL program which implements the algorithm.

(Hevner and Yao, 1987) have demonstrates current directions taken to apply query optimization to distributed database systems on local area networks. The query optimization algorithms are adapted to the unique aspects of the local area network environment. A classification taxonomy is presented and used to analyze the proposed query-optimization algorithms. Research on distributed query optimization can be classified based upon the distributed system environment, the optimization objectives, and the algorithmic techniques used to derive the final query execution strategy. A number of decisions must be made within each of these areas in order to define the context, objectives, and techniques of query optimization. The unique features of each algorithm are highlighted and a qualitative comparison of the algorithms.

(Morrissey and Osborn, 1998)have suggested an approach for general queries which uses reduction filter which are based on Bloom filters, to minimize data transfers and reduce local processing costs.

(Aljanaby et al. 2005) have suggested the major optimization issues being addressed in distributed databases. Query optimizer is mainly consists of three components: The search space, The search strategy, The cost model. The optimizer enumerates alternative plans, estimates the cost of every plan using a cost model and chooses the plan with lowest cost.

(Pournaghshband and Movafaghi, 2007)have first discussed the major issues regarding query plan evaluation for query processing and showed how the query optimizer can choose an optimal plan for efficient execution of queries. Second, they discussed an efficient process for designing an optimizer which examines frequently used queries and identifies two categories of group of queries. One group of queries requiring the same procedure and one group of queries requiring data from the same site for producing the result. Finally, they presented an algorithm which examines frequently used queries and identifies those two group of queries.

(Sun et al. 2008) have suggested the skyline-join operator into relational database systems. The new operator skyline-join, as it is a hybrid of skyline and join operations. In this propose two efficient approaches to process skyline-join queries which can significantly reduce the communication cost and processing time.

(Hannaford et al., 2009) have presented a ARRQ technique to process queries with a minimum quantity of intersite data transfer. The technique can be used to process the query where all of the relations referenced by a query are non fragmented but distributed in different sites. The proposed technique is used to determine which relations are to be partitioned into fragments, and where the fragments are to be sent for processing. The technique is efficient compared to other techniques, as it generally chooses more than one relation to remain fragmented which exploits parallelism, while replicating the other relations (excluding the fragmented relations) to the sites of the fragmented relations. Thus the communication costs and local processing costs can be reduced due to the reduced size of the fragmented relations and the response time of queries can be improved.

(Hongxia and Weifeng, 2009) have suggested a Distributed Database Searching System model with the grid computing architecture Alchemi which is based on .NET**.** Alchemi is an easy-to-use .Net Grid-computing framework that aims to lower the barrier of entry into the world of Grid computing. The system that developed by Alchemi is not only easy to configure, but also able to assure the efficiency and accuracy of the query.

(Yuanyuan and Xifeng, 2010) have suggested a new algorithm in distributed database system query optimization , and this algorithm can significantly reduce the amount of intermediate result data, effectively reduce the network communication cost, to improve the optimization efficiency.

(Jing-Min and Guo-Hui, 2010) have suggested a Hadoop based distributed database system model. Apache Hadoop is a distributed, open source, high reliability and scalability computing platforms, which can be composed of a large number of low-cost hardware devices and run applications on the cluster. It can greatly reduce the cost of inputs, and provides a feasible method for the cloud computing application system.

(Sukheja and Singh, 2011) have presented approach of query optimization is very useful for distributed database systems. To computations of cost from the optimization process, the optimizer must consult the data sources involved in an operation to find the cost of that operation. A novel optimization approach in the distributed database environment, somewhat unexpectedly, indicate that a simple two-phase optimization scheme performs fairly well as long as the physical database design is known to the optimizer, though more determined algorithms are required.

(Kumar et.al, 2011) have proposed a novel method for query optimization using heuristic based approach to evaluate the efficiency of a query search in the database operations. In the proposed algorithm, a query is searched using the storage file which shows an improvement with respect to the earlier query optimization techniques. Also, the improvement increases once the query goes more complicated and for nesting query. Therefore, heuristic based query optimization is a better approach to query optimization as compared to earlier query optimization techniques.

(Thakare et.al, 2011) have elaborated three types of algorithms. First, deterministic algorithm, namely the exhaustive search dynamic programming algorithm. It produces optimal left-deep processing trees with the big disadvantage of having an exponential running time. This means, that for queries with more than 10-15 joins, the running time explodes. Genetic and randomized algorithms on the other hand don’t generally produce an optimal access plan. But in exchange they are superior to dynamic programming in terms of running time. They have chosen a better join ordering the temporarily stored relation contains only one row. So consider the great importance of the join ordering for minimizing the number of rows. Iterative Improvement algorithms have shown that it is possible to reach very similar results with randomized algorithms depending on the chosen parameters.

(Sharma et.al, 2012) have computing and analyzing the performance of joins and semi joins in distributed database system. The various metrics considered while analyzing performance of join and semi join in distributed database system are Query Cost, Memory used, CPU Cost, Input Output Cost, Sort Operations, Data Transmission, Total Time and Response Time. The data transmission in a distributed query using semi join is always lesser than the data transmitted in distributed query using joins operation however data accessed using semi join may be larger than join operation. semi joins implement more operation as compare to join, but it reduces the number of bytes transferred from one site to another to great extent. Further one is able to conclude that semi joins are beneficial if the transmission cost is of main consideration, otherwise joins will be preferred.

(Chen et.al, 2012) have analyzed the query optimization process based on semi-join operation combined with the practical application. In addition, it introduced a classical algorithm which is used for multiple connection and query optimization based on the semi-join query optimization, the SDD-1 algorithm.

(Mor et al. 2012) have analyzed different techniques of query optimization in relational databases and compared their performance. Different techniques using different representations show that there are many other ways to represent query other than the query trees. The query graph is simple method to represent the query. Tableaus on the other hand are simple to define, use and implement. Graphs and tableaus provide a very easy way to optimize the query in databases and the greedy method used for optimization of single block queries can be extended to optimize the multi block queries.

(Mahajan and Jadhav, 2013) have suggested the Bloom join as join algorithm for query optimization which uses the hash function and reduces the total processing cost. It works better than semi-join and reduces transmission cost. Bloom join with open source map-reduce framework of hadoop improves the performance of query optimization. Hadoop is a data grid operating system which provides an economically scalable solution for storing and processing large amounts of unstructured or structured data over long periods of time. They have used the reduce side join with bloom filters which is inexpensive than map-side join.

PROPOSED WORK

The optimization algorithm is based on relational algebra equivalence transformation [Yuanyuan and Xifeng, 2010]. The basic idea of **Proposed algorithm** is: To convert the query problem into relational algebra expression, analyze the obtained query syntax tree, and optimize according to equivalence rules.

The algorithm first uses relational algebra equivalence transformation to raise the connecting and merging operations in the query tree as much as possible, while moving the selection and projection operations down to the fragment definition. Then to determine the horizontal or vertical slice, if the horizontal slice, to compare with the slice and the selection conditions to remove the contradictory fragments, if only a fragment left then to remove one parallel operation. If the vertical slice, then to compare with the fragment property set and the attribute set involved in projection operation to remove all the unrelated fragments. If only one vertical segment left then to remove one connection operation, thus to achieve the purpose of optimization query.

It is to use heuristic optimization method to optimize the relational algebra expression. And in the relational algebra expression, the most time and space operation is Cartesian product and join operations, so, the implementation of selection and projection operations as early as possible, to avoid the direct Cartesian product operation, then combines a series of selection and projection before and after it together to reduce the size of intermediate relations, thus to achieve optimization.

The Proposed algorithm utilizes some of the rules of transform an initial query tree into an optimized tree that is efficient to execute during heuristic optimization.

INPUT: Relational algebra query
OUTPUT: Best evaluation plan
Algorithm:

Step 1: Design the initial canonical tree of the query.
Step 2: Move the SELECT Operation down the query tree.
Step 3: Apply more restrictive SELECT operation first. If the two relations are residing at same site, they will be handled first.
Step 4: Replace CARTESIAN PRODUCT and SELECT with JOIN operation.
Step 5: Move PROJECT operations down the query tree.

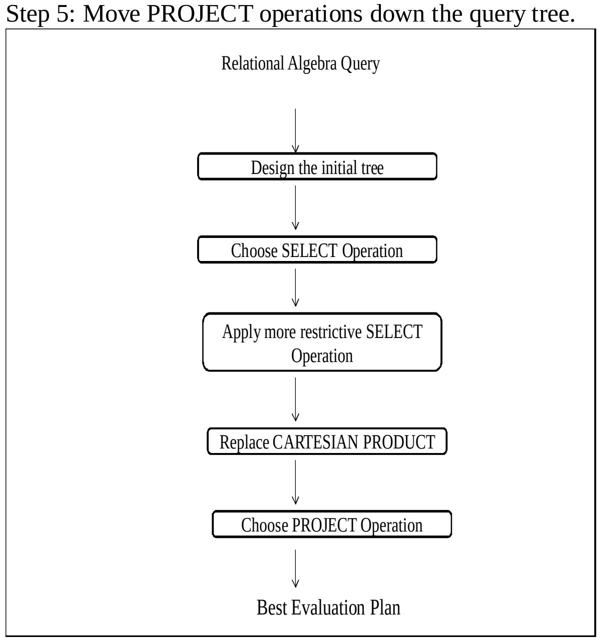


Fig.1: Heuristic Query Optimization

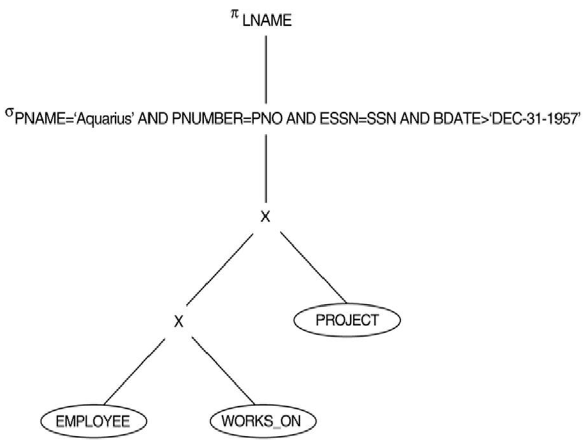
Steps in Converting a query tree during heuristic optimization

There are following steps in converting a query tree during heuristic optimization by using various rules. First we have query in SQL and convert it into relational algebra query.

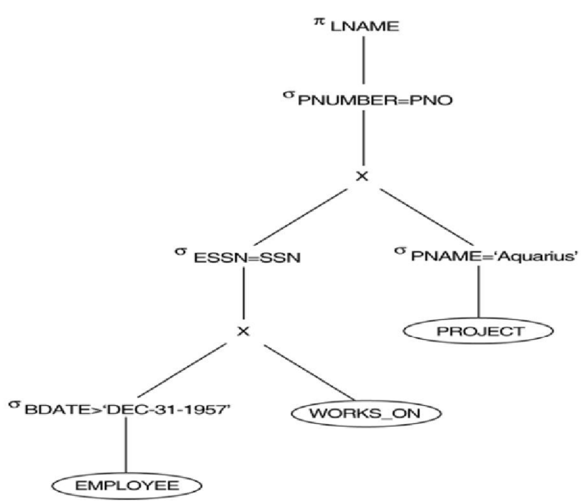
Example: Select names of employees working on the 'Aquarius' Project and born after 1957.

Select Lname from employee, works_on, project where pname='acquarius' and birthdate > '12/31/1957' and ssn=essn and pnumber=pno.

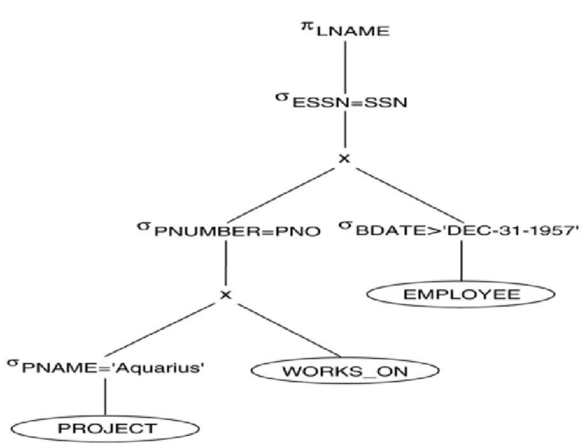
A. Initial query tree for the SQL query made by parser



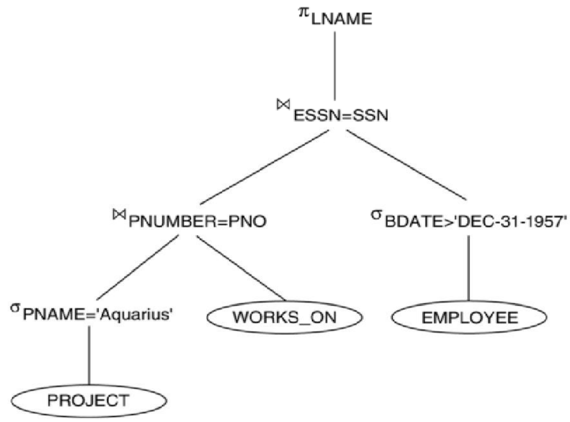
B. Moving SELECT operations down the query tree



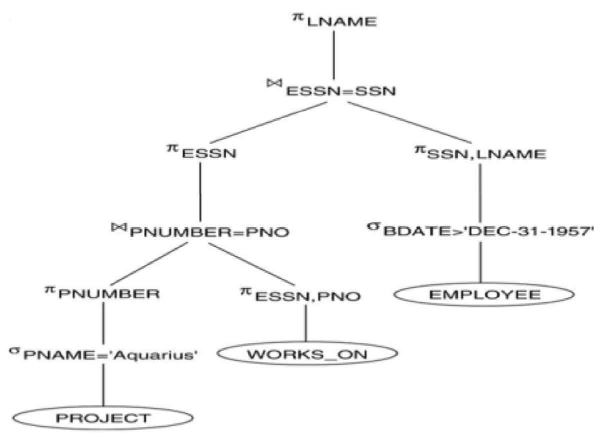
C. Applying the more restrictive SELECT operation first



D. Replacing CARTESIAN PRODUCT and SELECT with JOIN operations



E. Moving PROJECT operations down the query tree



EXPERIMENTAL ANALYSIS

The Proposed Query Optimization algorithm is implemented in visual studio 2010 as front end and SQL Server 2005 as back end. In this framework, the code is written in which to find the cost of different operations like Cartesian product, semi-join, and join operation. To evaluate the overall communication cost for each operation of relational algebra and also to evaluate how many number of records access for sites like considered three sites of site1, site2 and site3.

In SQL Server 2005, consider a database of data_allocation having the following relation schemas:

Cost_calc (site, cost, threshold, counter)

Cost_table (site_name, semijoin_cost, join_cost, Cartesian product_cost, communication_cost, record_count, site_frequency)

Master_table (site_name, thresh_hold, counter, request_from)

Parameter_table (record_count, site1, site2, site3)
Site1 (st_rollno, st_name)
Site2 (st_rollno, st_name)
Site3 (st_rollno, st_name)
Student (st_rollno, st_name, course_code, class_code, session, semester).

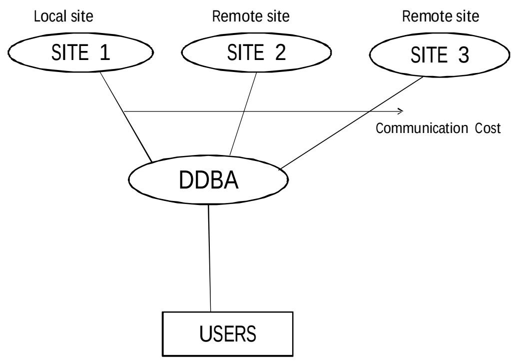


Fig.2**:** Flowchart of DDBA

In the Figure 2, assume of three sites eg. site1, site2, site3. Site 1 is located at local and site2& site3 are located at remote site.

The DDBA system receives the query from user those query access of user if we had the following query:

SELECT * FROM STUDENT WHERE ST_ROLLNO > _ AND ST_ROLLNO < _.

Then DDBA check where the operation performed whether the site1, site2 or site3. Data will be checked according to those operations. Then it will check how many times number of records counted and all those counts will be included in communication cost. Here, the communication cost will be static whereas, the record cost will be dynamic.

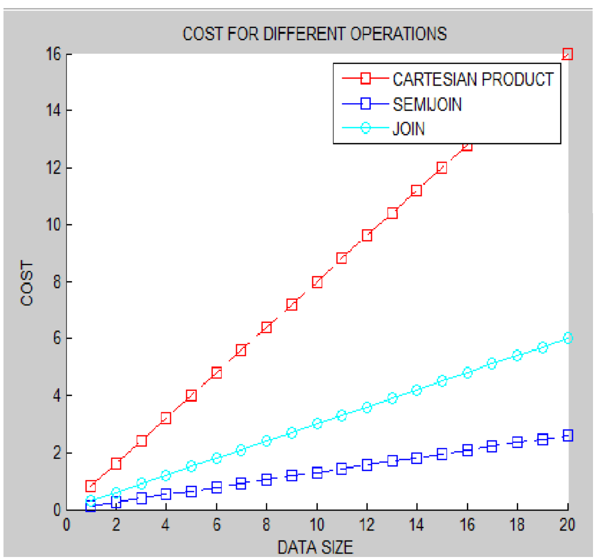


Fig.3: Cost for different operations

Figure 3, shows the cost of different operations. In this graph we compared cost of three different operations. Firstly, we compare of Cartesian product with join operation and then compared of join operation with semi-join.

This graph shows that ***Semi-join*** is better operation because it reduces the cost of joins.

When data size increases then cost of semi-join operation decreases.

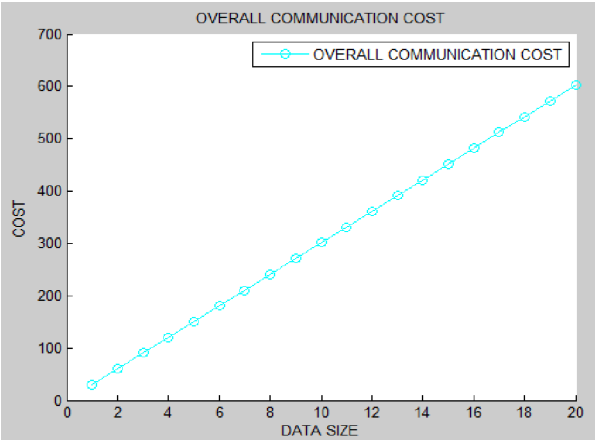


Fig.4**:** Overall Communication Cost

Figure 4, shows the graph of overall communication cost. The communication cost is to find by using these formulas:

Comm. Cost = record count * cost_ per_ site;

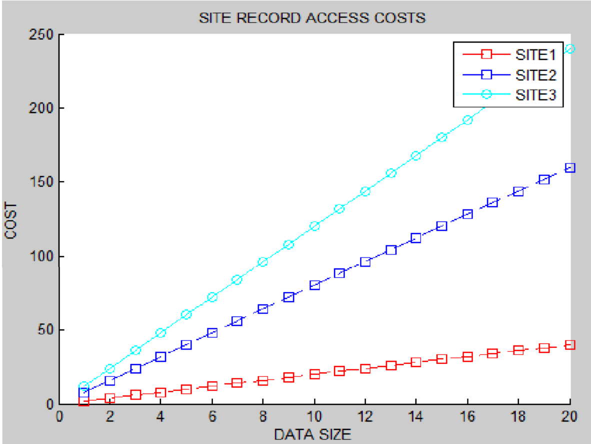


Fig.5: Site Record Access Costs

Figure 5, shows the graph of how many number of records access of sites using different operations. The overall cost of reduce the amount of data access.

CONCLUSIONS

In this paper, I have proposed query optimization algorithm is based on relational algebra equivalence transformation. The heuristic approach is used for selecting the optimal evaluation plan and semi-join approach for reducing the communication cost. The main objective is to reduce the cost of query optimization algorithm. The communication cost is one of the key factors among them. To evaluate the cost of using different operations of relational algebra. The implementation of selection and projection operations in the relational algebra expression, to avoid the direct Cartesian product operation, then combines a series of selection and projection before and after it together to reduce the size of intermediate relations, thus to achieve optimization. The Simulation results indicate that in case of *semi-join*, the cost is increasing at lower rate, in case of *join*, the cost is increasing more than that of semi-join whereas the Cartesian product operation shows the straight line when data size is increases, the cost is also increases. Hence, Conclude that semi-join is more preferable when the table data is more and it also reduces the communication cost.

References

[1] Aljanaby.A, Abuelrub.E and Odeh.M, (2005) *"A survey of distributed query optimization"*, International Arab Journal of Information Technology (IAJIT), Jordan, vol. 2, no. 1, pp. 48-57.

[2] Bernstein.P and Chiu.D (1981) "*using semi-joins to solve relational queries*" journal of the association for computing machinery vol.28, no.1, pp.25-40.

[3] Chung.C and Irani.K, (1986) "*An optimization of queries in distributed database systems*" journal of parallel and distributed computing pp.137-157.

[4] Chen.Y, Zhou.L, Li.T and Yu.Y, (2012) "*The semi-join query optimization in distributed database system*" National Conference on Information Technology and Computer Science (CITCS), pp. 606-609.

[5] haemi R. and Milanifard A. (2008) "*Evolutionary Query Optimization for Heterogeneous Distributed Database Systems*", World Academy of Science, Engineering and Technology 43.

[6] Hevner.A Apers.P and Yao.S (1983) "*optimization algorithms for distributed queries*" in proceedings of IEEE transactions on software engineering vol. SE-9, No.1. pp. 57-68.

[7] Hevner.A and Yao.S, (1987) "*Querying distributed databases on local area networks*" in proceedings of the IEEE conference, vol. 75, no. 5,pp. 563-572.

[8] Hannaford.M, Henskens.F and Alom.B, (2009) "*Query processing and optimization in distributed database systems*" international journal of computer science and network security (IJCSNS) vol.9, no.9, pp.143-152.

[9] Hongxia X. and Weifeng L. (2009) "*Distributed Database Searching System based on Alchemi*" in proceedings of the 9th international conference on IFCSTA volume01 pp. 160-163 ISBN: 978-0-7695-3930-0

[10] Jing-Min L. and Guo-Hui H. (2010) "*Research of Distributed Database System Based on Hadoop*" in proceedings of the 2nd international conference on ICISI pp. 1417-1420 ISBN 978-1-4244-7616-9.

[11] Kumar.S, Khandelwal.G, Varshney.A and Arora.M, (2011) "*cost based query optimization with heuristics*" international journal of scientific and engineering research vol.2, pp.1-6, ISSN-2219-5518.

[12] Morrissey J. and Osborn W. (1998) "*Distributed query optimization using reduction filters*" in Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering volume 02 pp. 707-710 ISBN 0-7803-4314-X.

[13] Mor.j, Kashyap.I and Rathy.R, (2012) "*Analysis of query optimization techniques in databases*" International Journal of Computer Applications *Volume 47* pp.5-9.

[14] Mahajan.S and Jadhav.V, (2013) "*bloom join fine-tunes distributed query in hadoop environment*" international journal of advanced engineering technology (IJAET) vol.4, pp.67-71, E-ISSN - 0976-3945.

[15] Ozsu M. and Valduriez P. (1991) Chapter 1 Introduction, "*Principles of Distributed Database Third Edition*" 3rd edition, Springer, 2011; ISBN978-1-4419-8833-1.

[16] Oszu, M. T. and Valduriez P.(1997) "*Distributed and Parallel Database Systems*," in Trucker A. (Ed), The Computer Science and Engineering Handbook, CRC press, pp. 1093-1111.

[17] Pournaghshband.V and Movafaghi.S (2007) "*An optimal query execution plan for database systems*" in proceedings of IRMA international conference pp. 648-649.

[18] Sun.D, Wu..S, Li.J and Tung.A, (2008) "*Skyline-Join in Distributed Databases*" in proceedings of the 24th international conference on ICDEW pp. 176-181 ISBN 978-1-4244-2162-6.

[19] Sukheja D. and Singh U., (2011) "*A Novel Approach of query optimization for distributed database systems*" international journal of computer science issues (IJCSI) vol.8, issue 4, no 1, pp.307-312.

[20] Talbot.S, (1984) "*An investigation into logical optimization of relational query languages*" the computer journal, vol. 27, no. 4, pp. 301-309.

[21] Thakare.P, Jadhao.S and Pund.M, (2011) "*A role of query optimization in relational database*" International Journal of Scientific & Engineering Research, Volume 2, Issue 1,pp. 1-5,ISSN 2229-5518.

[22] Virk.R, Singh.G and Sharma.M, (2012) "*Analysis of joins and semi joins in a distributed database query*" international journal of computer applications, volume 49, no.16, pp.14-18.

[23] Yuanyuan F. and Xifeng M. (2010) " *Distributed Database system query optimization algorithm research*" in proceedings of the 3rd international conference on ICCSIT volume8 pp. 657-660 ISBN 978-1-4244-5537-9.